

Федеральное государственное автономное
образовательное учреждение
высшего образования
"СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ"

Космических и информационных технологий
институт

Информационные системы
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой ИС
_____ С. А. Виденин
подпись инициалы, фамилия
« ____ » _____ 2017 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 "Информатика и вычислительная техника"
код и наименование специализации
Разработка web-приложения "Органайзер"
тема

Руководитель _____
подпись, дата

Л. С. Троценко
инициалы, фамилия

Выпускник _____
подпись, дата

Д. А. Мухортиков
инициалы, фамилия

Нормоконтролер _____
подпись, дата

Л. С. Троценко
инициалы, фамилия

Красноярск 2017

РЕФЕРАТ

Выпускная квалификационная работа по теме "Разработка web-приложение Органайзер" содержит 43 страниц текстового документа, 2 таблицы, 13 изображений, 6 использованных источников.

ПРОЕКТИРОВАНИЕ, WEB-ПРИЛОЖЕНИЕ, УП2, ОРГАНАЙЗЕР, САЙТ, HTML, КЛИЕНТ-СЕРВЕР

Объект исследования: Интернет технологии.

Предмет исследования: планировщик задач и расписаний.

Целью данной выпускной квалификационной работы является создание инструмента для быстрой и удобной организации личного и рабочего времени.

Задачи ВКР.

а) Анализ приложений на рынке, реализованных для решения задачи оптимизации личного и рабочего времени.

б) Проектирование функционала web-приложения, разработка алгоритмов реализации функционала приложения.

в) Разработка интуитивно понятного пользовательского интерфейса, для удобной работы с web-приложением.

В результате выполнения ВКР разработано web-приложение "Органайзер", позволяющее с меньшими трудозатратами составить план работы на день и поставить необходимые задачи для себя и своих сотрудников.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 Анализ архитектуры системы	7
1.1 Анализ аналогов приложения	7
1.1.1 1С:Дистрибьюция. LeaderTask	7
1.1.2 JIRA	7
1.2 Анализ технологий	8
1.2.1 Web - приложение	8
1.2.2 Клиент - серверное приложение	10
1.2.3 Model - View - Controller	10
1.3 Выбор программного средства для разработки	11
1.3.1 HTML	11
1.3.2 CSS	12
1.3.3 Фреймворк Bootstrap 3	13
1.3.4 JavaScript	15
1.3.5 JQuery	17
1.3.6 PHP	17
1.3.7 Фреймворк Yii2	18
2 Проектирование и разработка web - приложения "Органайзер"	19
2.1.1 Моделирование web - приложения "Органайзер"	19
2.1.2 Проектирование функционала web-приложения "Органайзер"	20
2.1.3 Схема базы данных web-приложения "Органайзер"	23
2.2 Технологии разработки web-приложения "Органайзер"	24
2.2.1 Nginx web-сервер	24
2.2.2 Vagrant	24
2.2.3 Работа с базами данных	26
2.2.4 Генерация кода средством Gii	27
2.2.5 Работа с фреймворком Bootstrap 3	28

2.3	Режимы работы web-приложения "Органайзер"	30
2.3.1	Регистрация пользователей	30
2.3.2	Работа с компаниями	32
2.3.3	Работа с задачами	36
2.3.4	Работа с календарём	38
2.3.5	Распределение возможностей и привилегий	40
ЗАКЛЮЧЕНИЕ		42
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ		43

ВВЕДЕНИЕ

Web-приложение "Органайзер" создаётся для оптимизации трудозатрат связанных с организацией личного и рабочего времени.

Многие люди часто сталкиваются с проблемой лишних временных затрат при организации своего времени. Будь то директор компании, которому приходится затрачивать немало времени на постановку задач для сотрудников, назначение времени встреч и сроков проведения различных мероприятий или человек, занимающийся какой-либо частной деятельностью, для которого так же важно иметь возможность легко и быстро обозначить список необходимых задач и выделить время под необходимые мероприятия.

Данное приложение предназначено как для частного, так и корпоративного использования. Приложение даёт возможность быстро и легко сформировать список задач, распределить задачи по сотрудникам. Назначить время проведения встреч и мероприятий. Удобный интерфейс приложения позволит, имея лишь доступ в Интернет, всегда получить доступ к наглядному плану работы на день.

Руководителям часто приходится индивидуально прорабатывать задачи для каждого сотрудника. Тратить личное время, что бы довести информацию о задачах и встречах до своих работников. Необходимость в этом пропадает с web-приложением "Органайзер".

Целью выпускной квалификационной работы является создание web приложения для сокращения временных ресурсов связанных с организацией рабочего процесса компании или организации личных дел.

Для реализации поставленной цели необходимо решение следующих задач:

- анализ предлагаемых программных продуктов нацеленных на организацию личного и рабочего времени;
- проектирование функционала web-приложения;
- разработка дизайна веб приложения;
- разработка алгоритмов реализации функционала web-приложения;

- проектирование базы данных;
- реализация адаптивного, интуитивно понятного интерфейса.

1 Анализ архитектуры системы

1.1 Анализ аналогов приложения

На рынке существует большое количество программных продуктов направленных на упрощение выполнения задач, связанных с организацией времени. Ниже рассмотрим некоторые из них.

1.1.1 1С:Дистрибьюция. LeaderTask

Популярный деловой органайзер. Нацелен на планирование встреч, задач, управление проектами, а так же обработки сообщений электронной почты.

Основные функции:

- список дел;
- список задач;
- календарь встреч;
- список контактов;
- почтовый клиент.

У LeaderTask большой функционал направленный на решения множества задач. Основным недостатком данного продукта является его платная платформа 1С.

1.1.2 JIRA

Мощный онлайн сервис, позволяющий руководителям компаний планировать проекты, назначать исполнителей на задачи, отслеживать их выполнение. Основной функционал сервиса узконаправлен, и нацелен на работу с командой разработчиков. Сервис не подходит для индивидуального использования в личных целях. Из недостатков JIRA можно отметить большое количество настроек, на изучение которых потребуется дополнительное время.

При проектировании функционала web-приложения "Органайзер", основной акцент был сделан на интуитивно понятный интерфейс не перегруженный функционалом. Приложение нацелено как на частное использование в личных целях, так и на решение основных организационных моментов компаний.

1.2 Анализ технологий

1.2.1 Web - приложение

В наше время сфера информационных технологий имеет высокие темпы развития. В области программных продуктов происходит заметные изменения, всё большую популярность набирают web-приложения, которые развились из статичных сайтов, путём внедрения скриптов и наращивания большого объёма функционала. На данный момент web-приложения по функционалу ни чем не уступают десктоп-приложениям.

Далее приведена сравнительная таблица web-приложений и десктоп-приложений.

Таблица 1 - Сравнение web-приложений и десктоп-приложений

Критерий	Десктоп-приложение	Web-приложение
Доступ к сети Интернет	Не требуется.	Необходим. В качестве исключения некоторые приложения способные некоторое время работать в автономном режиме.
Установка и обновление	Необходима установка. Обновление при выходе новой версии.	Нет необходимости в установке и обновлении.

Продолжение таблицы 1

Критерий	Десктоп-приложение	Web-приложение
Интерфейс взаимодействия	Стандартные интерфейсы, стандартное взаимодействие.	Разнообразный интерфейс взаимодействия, разнообразные реализации.
Совместимость с устройствами	Зависимость от платформы.	Не зависит от платформы.
Анимация, графика	Быстрая анимация и отклик на действия.	В связи необходимости передачи данных по сети, относительно медленный отклик.
Поиск по контенту	Невозможно, кроме тех случаев когда это реализовано на уровне приложения.	Есть.
Разработка	Использование разных инструментов под разные платформы. Зачастую приходится писать свою версию под каждую версию платформы	Кроссплатформенно. Все выполняется на сервере. Необходим только браузер. Все инструменты и софт на сервере кроссплатформенны.

Реализация проекта в виде web-приложения позволит получить кроссплатформенный продукт с удобным доступом на любой платформе.

1.2.2 Клиент - серверное приложение

"Клиент - сервер" - сетевая архитектура. Задания в данной системе распределяются между серверами и клиентами. Программа - сервер ожидает от клиентских программ запросы и предоставляет им свои ресурсы в виде web-страниц, обработанных баз данных, сервисных функций и много другого.

Преимущества:

- отсутствие дублирования кода программы-сервера программами клиентами;
- все вычисления происходят на стороне сервера. Требования к компьютерам, на которых установлен клиент, очень низкие;
- все данные хранятся на сервере, как правило сервер защищён намного лучше большинства клиентов.

Клиент-серверное приложение позволяет намного проще реализовать контроль полномочий для клиентов с разным уровнем доступа.

1.2.3 Model - View - Controller

Model-View-Controller - MVC модель или так же модель - представление - контроллер. Данная модель разделяет данные приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента:

- модель - компонент системы отвечающий за извлечение пользовательских данных, с последующей их манипуляцией;
- представление - отвечает за отображение основной видимой структуры и данных на сайте;
- контроллер - осуществляет управление и контроль за работой моделей и представлений.

Логическая схема работы Model-View-Controller представлена на рисунке 1.

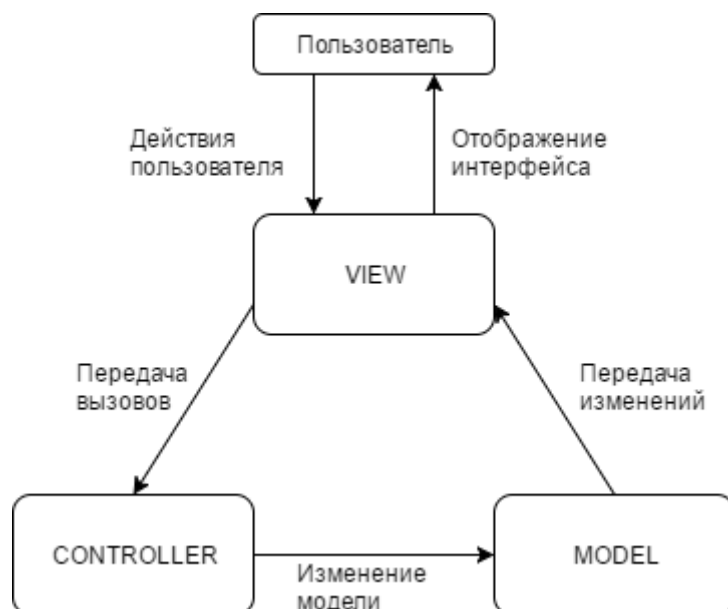


Рисунок 1 - Логическая схема работы Model-View-Controller

Использование Model-View-Controller даёт нам возможность изменять каждый компонент независимо друг от друга и чётко разграничить функционал и задачи каждого элемента приложения.

1.3 Выбор программного средства для разработки

1.3.1 HTML

Для создания, хранения и отображения гипертекста во Всемирной паутине традиционно используется язык HTML.

Язык HTML - язык тэгов. Теги формируют структуру HTML документа. Теги оформляются угловыми скобками, между которыми прописывается имя тега.

Теги располагаются в HTML-документе в соответствии с правилами разметки (порядок следования, правило вложенности тегов), создавая разделы будущей веб-страницы. Кроме того, HTML-документы могут содержать специальные символы.

Браузер просматривает (интерпретирует) HTML-документ, выстраивая его структуру и отображая ее в соответствии с инструкциями, включенными в этот файл (таблицы стилей, скрипты). Если разметка правильная, то в окне браузера будет отображена HTML-страница, содержащая HTML-элементы - заголовки, таблицы, изображения и т.д.

Процесс интерпретации начинается прежде, чем веб-страница полностью загружена в браузер. Браузеры обрабатывают HTML-документы последовательно, с самого начала, при этом обрабатывая CSS и соотнося таблицы стилей с элементами страницы.

1.3.2 CSS

CSS (Каскадные таблицы стилей) - формальный язык описания внешнего вида документа, написанного с использованием языка разметки.

Преимущественно используется как средство описания, оформления внешнего вида веб-страниц, написанных с помощью языков разметки HTML и XHTML, но может также применяться к любым XML-документам.

Каскадные таблицы стилей описывают правила отображения элементов с помощью свойств и допустимых значений этих свойств. Для каждого элемента можно использовать ограниченный набор свойств, остальные свойства не будут оказывать на него никакого влияния.

CSS используется создателями веб-страниц для задания цветов, шрифтов, расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц. Основной целью разработки CSS являлось разделение описания логической структуры веб-страницы (которое производится с помощью HTML или других языков разметки) от описания внешнего вида этой веб-страницы (которое теперь производится с помощью формального языка CSS). Такое разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом. Кроме того,

CSS позволяет представлять один и тот же документ в различных стилях или методах вывода, таких как экранное представление, печатное представление, чтение голосом (специальным голосовым браузером или программой чтения с экрана), или при выводе устройствами, использующими шрифт Брайля.

Объявление стиля состоит из двух частей: элемента веб-страницы - селектора, и команды форматирования - блока объявления. Селектор сообщает браузеру, какой именно элемент форматировать, а в блоке объявления (код в фигурных скобках) перечисляются формирующие команды - свойства и их значения.

1.3.3 Фреймворк Bootstrap 3

Bootstrap 3 это популярный фреймворк, содержащий свободный набор инструментов для разработки адаптивных сайтов и мобильных web-проектов. Включает в себя HTML и CSS шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и других компонентов веб-интерфейса, включая JavaScript-расширения.

Данный фреймворк начал разрабатываться в качестве внутренней библиотеки компании Twitter под названием Twitter Blueprint. После нескольких месяцев разработки он был открыт под названием Bootstrap 19 августа 2011 года.

Основными нововведениями второй версии, появившейся 31 января 2012 года, стали 12 колоночная сетка и поддержка адаптивности.

Третья версия была выпущена 19 августа 2013 года. В ней адаптивность получила дальнейшее развитие, был осуществлён переход к концепции mobile first, оптимизации прежде всего под мобильные устройства. Дизайн по умолчанию стал плоским.

Работа над четвёртой версией начата 29 октября 2014 года. Альфа версия вышла 19 августа 2015 года.

Основные преимущества Bootstrap 3:

- экономия времени - Bootstrap позволяет сэкономить время и усилия, используя шаблоны дизайна и классы, и сконцентрироваться на других разработках;
- высокая скорость - динамичные макеты Bootstrap масштабируются на разные устройства и разрешения экрана без каких-либо изменений в разметке;
- гармоничный дизайн - все компоненты платформы Bootstrap используют единый стиль и шаблоны с помощью центральной библиотеки.
- дизайн и макеты веб-страниц согласуются друг с другом;
- простота в использовании - платформа проста в использовании, пользователь с базовыми знаниями HTML и CSS может начать разработку с Twitter Bootstrap;
- совместимость с браузерами - Twitter Bootstrap совместим с Mozilla Firefox, Yandex Browser, Google Chrome, Safari, Internet Explorer, Microsoft Edge и Opera;
- открытое программное обеспечение - особенность Twitter Bootstrap, которая предполагает удобство использования, посредством открытости исходных кодов и бесплатной загрузки.

Bootstrap спроектирован для лучшей работы в новых браузерах, то есть старые браузеры не всегда могут правильно отображать стили, хотя полностью функциональны в визуализации определенных компонентов.

Далее приведена таблица поддержки браузерами Bootstrap 3.

Таблица 2 - Поддержка Bootstrap 3 браузерами

ОС	Chrome	Firefox	Internet Explorer	Opera	Safari
Android	+	-	N/A	-	N/A
iOS	+	N/A	N/A	-	+
MAC OS X	+	+	N/A	+	+
Windows	+	+	+	+	-

Основные инструменты Bootstrap 3:

- сетки - заранее заданные размеры колонок, которые можно сразу же использовать;
- шаблоны - фиксированный или резиновый шаблон документа;
- типографика - описания шрифтов, определение некоторых классов для шрифтов, таких как код, цитаты и т. п.;
- медиа - представляет некоторое управление изображениями и видео;
- таблицы - средства оформления таблиц, вплоть до добавления функциональности сортировки;
- формы - классы для оформления форм и некоторых событий, происходящих с ними;
- навигация - классы оформления для табов, вкладок, страничности, меню и тулбара;
- алерты - оформление диалоговых окон, подсказок и всплывающих окон.

Использование Bootstrap 3 позволит создать адаптивный сайт, который будет хорошо просматриваемый на всех устройствах.

1.3.4 JavaScript

JavaScript - язык сценариев, с его помощью можно создавать интерактивные html-документы, производить вычисления, выполнять проверку допустимости данных без обращения к серверу.

Сценарий JavaScript не компилируется в бинарный код, а интерпретируется и обрабатывается интерпретатором, встроенным в браузер. Браузер начинает выполнять код JavaScript сразу же, как только обнаружит его на странице. Каждая строка кода выполняется последовательно, переход к следующей строке производится только после выполнения предыдущей. Если в документе содержится несколько сценариев, то они будут исполняться в порядке их расположения в документе.

Возможности JavaScript:

- изменять страницу: писать текст, добавлять и удалять теги, менять стили элементов;
- реагировать на события: скрипт имеет возможность ожидать выполнения действия (клика мышки) и реагировать на это вызовом функции;
- выполнять запросы к серверу и загружать данные без перезагрузки страницы;
- устанавливать и считывать cookie, валидировать данные, выводить сообщения и многое другое.

JavaScript подключается напрямую в HTML-файл. Самый простой способ - это написать javascript-команды внутри тега `<script>` в теле страницы.

Когда браузер читает HTML-страницу, и видит `<script>` - он первым делом читает и выполняет код, а только потом продолжает читать страницу дальше.

JavaScript - объектно-ориентированный язык. Следует ознакомиться с тремя терминами.

- Объекты - говоря простым языком, объект (object) - это какой-либо предмет (окно браузера, формы и их части, например кнопки и текстовые окна). В JavaScript также имеется собственная группа встроенных объектов, к которым относятся массивы, данные и т.д.

- Метод (method) - это действия, которые может выполнять объект. Примерами методов являются открытие и закрытие окон, нажатие кнопок. Здесь речь идет о трех методах: `open ()`, `close ()` и `click ()`.

- Свойства - у всех объектов имеются свойства (properties).

Практически невозможно представить себе хороший сайт без динамических визуальных и функциональных эффектов.

Использование JavaScript позволят сделать web-приложение современным и динамичным.

1.3.5 JQuery

JQuery - одна из библиотек JavaScript. Основной функционал библиотеки направлен на взаимодействие HTML и JavaScript. JQuery даёт возможность легко и быстро получать доступ к любому элементу DOM, управлять его атрибутами и содержимым. Так же эта библиотека предоставляет удобные методы работы с Ajax запросами.

Возможности JQuery:

- движок кросс-браузерных CSS-селекторов Sizzle, выделившийся в отдельный проект;
- перемещение по дереву DOM, включая поддержку XPath как плагина;
- события;
- визуальные эффекты;
- ajax дополнения;
- javascript плагины и многое другое.

JQuery направлен на удобство и снижение трудозатрат при работе с JavaScript. Данная библиотека обеспечит стабильность и кроссбраузерность проекта. С её помощью возможно сделать то же самое что и с JavaScript, но потребуется меньше усилий.

1.3.6 PHP

PHP - один из самых распространённых языков программирования общего назначения, имеет открытый исходный код. Данный язык программирования создан специально для разработки в web направлении, код на языке PHP может внедряться непосредственно в сам HTML.

Основное отличие PHP от JavaScript заключается в том, что PHP скрипт обрабатывается на сервере, формирует HTML модель и предоставляет её

клиенту. Клиент получает только результат выполнения скрипта и никогда не сможет выяснить какой именно код его произвёл.

1.3.7 Фреймворк Yii2

Yii2 - это высокопроизводительный компонентный PHP фреймворк. Основной целью фреймворка является быстрая разработка современных сайтов с высокой функциональной нагрузкой. Yii2 очень универсален, и может использоваться при разработке любого типа приложения. Благодаря его компонентной структуре и отличной поддержке кэширования, фреймворк особенно подходит для разработки таких крупных проектов как веб-порталы, форумы, CMS системы и Интернет магазины.

Сравнение Yii2 с другими фреймворками:

- как и многие другие PHP фреймворки, для организации кода Yii2 использует архитектурный паттерн MVC (Model-View-Controller);
- Yii2 придерживается философии простого и элегантного кода не пытаясь усложнять дизайн только ради следования каким-либо шаблонам проектирования;
- Yii2 является full-stack фреймворком и включает в себя проверенные и хорошо зарекомендовавшие себя возможности, такие как ActiveRecord для реляционных и NoSQL баз данных, поддержку REST API, многоуровневое кэширование и другие;
- Yii2 отлично расширяем. Вы можете настроить или заменить практически любую часть основного кода. Используя архитектуру расширений легко делиться кодом или использовать код сообщества;
- одна из главных целей Yii2 - производительность.

Разработка веб приложения, используя средства фреймворка Yii2, позволяют намного быстрее реализовать весь функционал web-приложения и создать конкурентно способный продукт.

2 Проектирование и разработка web - приложения "Органайзер"

Web-приложение "Органайзер" должен решать следующие задачи:

- регистрация пользователей;
- создание личных и корпоративных задач;
- назначение личных и корпоративных встреч;
- регистрация компаний, сотрудников а так же секретарей закреплённых за этими сотрудниками;
- распределение корпоративных задач между сотрудниками.

2.1.1 Моделирование web - приложения "Органайзер"

Структурная схема web-приложения "Органайзер" представлена на рисунке 2.

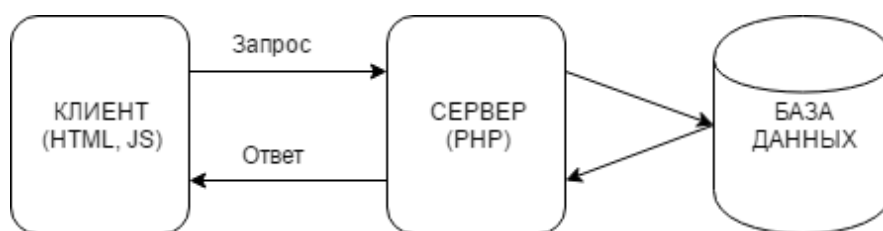


Рисунок 2 - Структурная схема web-приложения "Органайзер"

Web-приложение "Органайзер" состоит из клиентской и серверной части. Основные компоненты системы:

- web страницы (HTML), выполненные с использованием фреймворка Bootstrap 3, отвечают за отображение пользователю структуры и данных web-приложения;
- javascript файлы, с использованием которых реализовано динамическое изменение web-приложения;

- скрипт-файлы PHP, с помощью которых осуществляется контроль и передача данных между серверной частью, базами данных и клиентской частью.

2.1.2 Проектирование функционала web-приложения "Органайзер"

Функционал web-приложения "Органайзер" делится на три основных направления:

- работа с задачами - создание новых задач, удаление и изменение задач, назначение исполнителя и сроков выполнения задачи;
- работа с событиями - создание нового события в календаре на рабочем столе web-приложения, просмотр и удаление события;
- работа с сотрудниками компании - при регистрации компании в web-приложении "Органайзер", у директора компании появляется возможность привязывать другие аккаунты в качестве работников, в последствии назначать для них задачи. Так же есть возможность привязывать другие аккаунты в качестве секретарей.

В приложении предусмотрено 4 типа учётных записей:

- индивидуальный - учётная запись для индивидуального использования, на рабочем столе есть личный календарь и список задач, в главном меню присутствуют пункты для регистрации собственной компании и просмотра предложений о сотрудничестве в других компаниях в качестве работника или секретаря. Индивидуальный аккаунт является стартовым, любой пользователь, прошедший регистрацию в web-приложении, получает аккаунт с статусом "индивидуальный";
- директор - статус учётной записи изменяется с "индивидуальный" на "директор" при условии, что пользователь работающий под этим аккаунтом произведёт регистрацию своей компании в web-приложении. В главном меню перестанет отображаться пункт "предложения" и появится новый пункт "компания" в котором пользователь может просматривать список сотрудников

своей компании, при необходимости отправлять предложения о работе для новых сотрудников или увольнять старых. На рабочем столе пользователя всё так же отображается календарь с событиями, и список задач, но теперь присутствует функция назначения исполнителя для задачи, которого можно выбрать из списка сотрудников компании. Пользователь аккаунта со статусом "директор", так же может закрепить за собой личного секретаря, который будет иметь доступ к календарю событий директора и иметь возможность создавать и удалять события;

- работник - аккаунту присваивается статус "работник" при условии, что пользователь аккаунта зайдёт во вкладку предложений о сотрудничестве и примет предложения от одной из компаний на должность работника. При использовании аккаунта "работник" в главном меню перестают отображаться пункты меню для регистрации собственной компании или просмотра предложений о сотрудничестве, вместо них появляется пункт меню "компания", в котором пользователь имеет возможность закрепить за собой своего личного секретаря, отправив запрос на работу для какой-либо учётной записи. На рабочем столе пользователя отображается календарь с событиями, список составленных задач и список задач, для которых директор компании назначил его исполнителем;

- секретарь - статус учётной записи меняется на "секретарь" при условии, когда пользователь зайдёт во вкладку предложений о сотрудничестве и примет предложение от одной из компаний на должность секретаря. У аккаунта со статусом "секретарь" в главном меню пропадают пункты позволяющие зарегистрировать в приложении собственную компанию или принять предложение о сотрудничестве от других компаний. На рабочем столе секретаря отображается календарь непосредственного начальника, то есть пользователя за которым закреплён данный секретарь. Пользователь имеет доступ к редактированию событий, созданию новых событий и их удалению.

Функциональная схема web-приложения "Органайзер" представлена на рисунке 3.

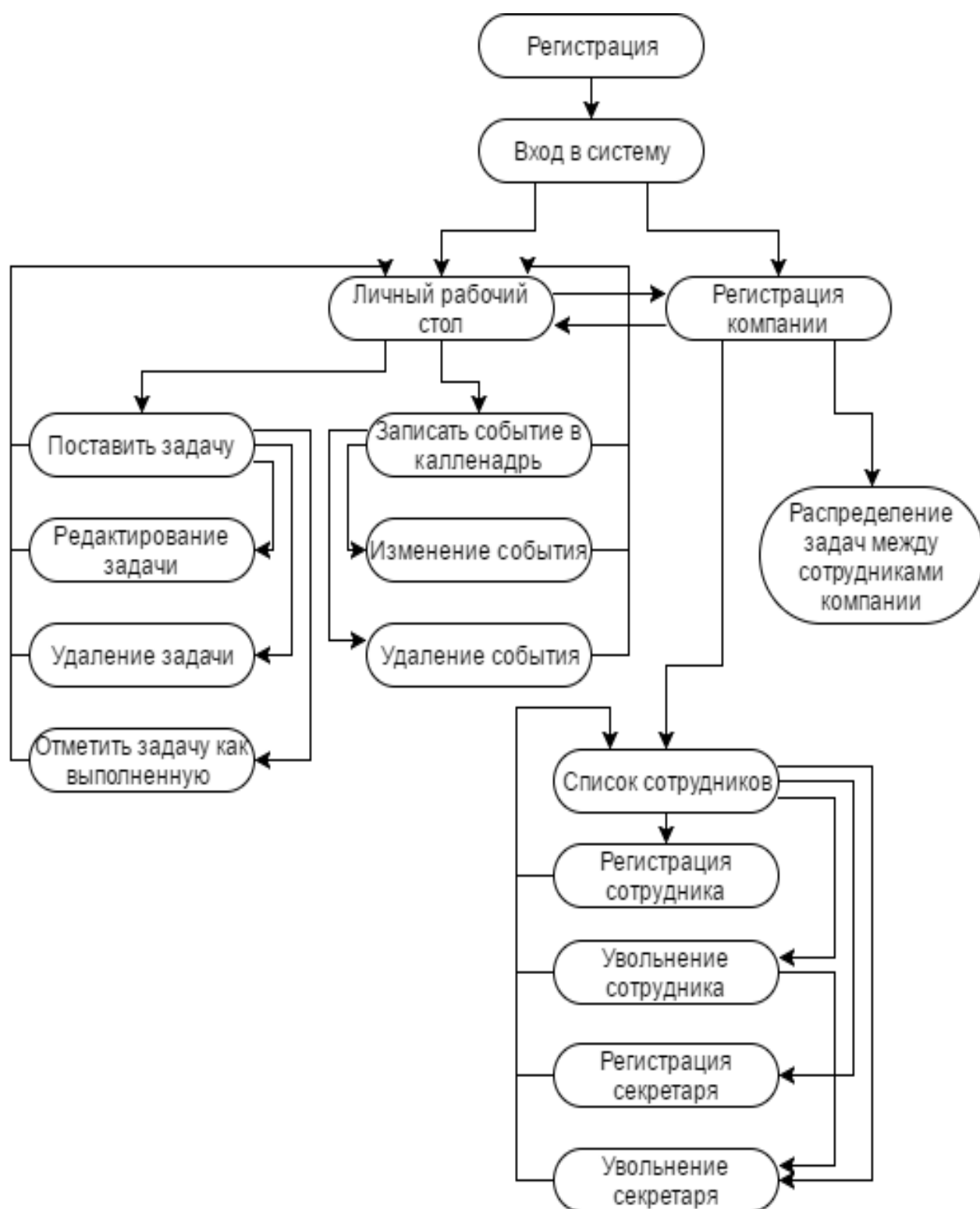


Рисунок 3 - Функциональная схема web-приложения "Органайзер"

2.1.3 Схема базы данных web-приложения "Органайзер"

Схема базы данных web-приложения "Органайзер" представлена на рисунке 4.

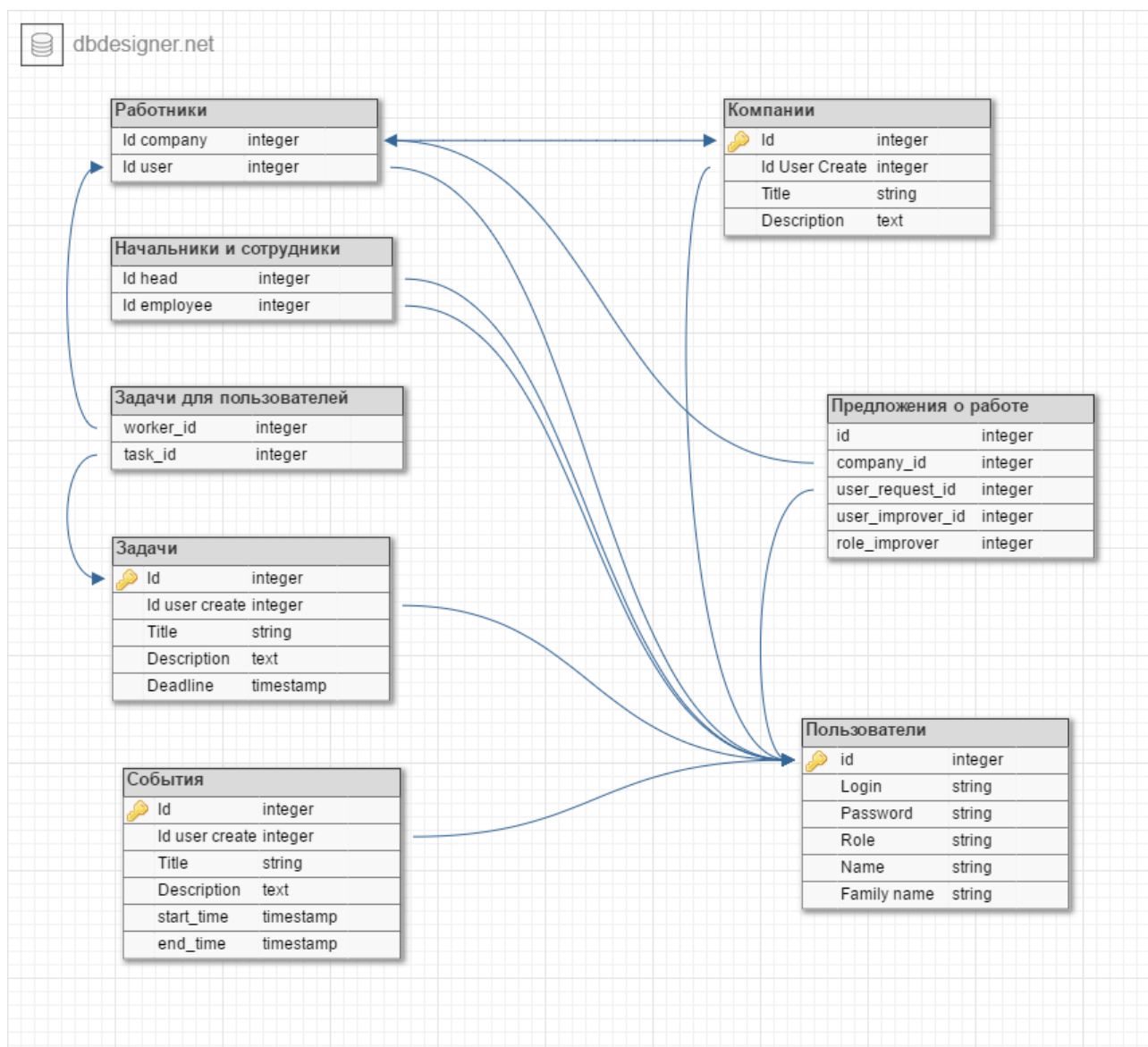


Рисунок 4 - Схема базы данных web-приложения "Органайзер"

2.2 Технологии разработки web-приложения "Органайзер"

Web-приложение состоит из двух основных частей. Серверной части, на которой обрабатываются все запросы в базу данных и клиентской части, отвечающей за отображение пользователю структуры приложения и запрошенных им данных.

2.2.1 Nginx web-сервер

Nginx web-сервер - самостоятельный HTTP-сервер. Данный web-сервер хорошо подходит для высоконагруженных проектов.

Возможности Nginx web-сервера:

- обслуживание статических запросов, индексных файлов, автоматическое создание списка файлов;
- распределение нагрузки и отказоустойчивость;
- модульность, в том числе сжатие (gzip), докачка (byte-ranges), преобразование изображений;
- поддержка HTTP/2 с приоритезацией на основе весов и зависимостей;
- настройка форматов логов, быстрая ротация логов;
- специальные страницы для ошибок;
- ограничение доступа в зависимости от адреса клиента.

При разработке web-приложения органайзер Nginx использовался как основной web-сервер. Для ускорения процесса разработки для настройки конфигурации Nginx использовался Vagrant.

2.2.2 Vagrant

Vagrant - средство позволяющее значительно проще и быстрее создать новую виртуальную машину. Vagrant является конфигурационным файлом, который автоматически разворачивается во внутреннем окружении новой

виртуальной машины. После чего появляется возможность с сетевого интерфейса виртуальной машины перенаправить порты и выделить IP-адрес, а так же установить необходимое программное обеспечение. Все настройки указываются в Vagrant файле.

Настройка Vagrant файла началась с указания локальных доменных имен, для этого в файле прописывался следующий код:

```
domains = {  
  backend: 'backend.organizer.dev',  
  frontend: 'organizer.dev',  
}
```

После чего указываются ссылки на файлы конфигураций для виртуальной машины.

```
config = {  
  local: './vagrant/config/vagrant-local.yml',  
  example: './vagrant/config/vagrant-local.example.yml'  
}
```

В файлах конфигураций были прописаны основные настройки виртуальной машины:

- GitHub Token уникальный ключ разработчика, без наличия которого невозможно вести разработку проекта на платформе Yii2;
- настройка серверного времени;
- проверки на обновления;
- имя виртуальной машины;
- ip-адрес виртуальной машины;
- количество процессорных ядер виртуальной машины;
- оперативная память виртуальной машины.

```
github_token: 442b21b09dec39da5794a2b68c5fb3faf9530628
```

```
timezone: UTC
```

```
box_check_update: false
```

```
machine_name: organizer
```

```
ip: 192.168.83.138
```

cpu: 1

memory: 512

После запуска команды Vagrant UP будет создана новая виртуальная машина с необходимыми параметрами и программным обеспечением.

2.2.3 Работа с базами данных

Для работы с базами данных при разработке web-приложения "Органайзер" на базе фреймворка Yii2, использовались миграции. Миграции баз данных позволяют как создавать новые таблицы с настройкой всех необходимых связей, так и изменять уже существующие таблицы, добавлять новые поля и редактировать связи между таблицами в базе данных.

Для создания новой миграции используется команда:

```
php yii migrate/create <name>
```

Команда прописывается в консоли, после чего создаётся файл с миграцией.

Для выполнения миграций используется команда:

```
php yii migrate
```

Информация о всех выполненных миграциях записывается в отдельную таблицу в базе данных.

Файл миграций содержит в себе код отвечающий за создание таблиц, их удаление или внесение изменений в уже существующую таблицу.

Для создания в базе данных таблицы содержащей данные о событиях и мероприятиях отображающихся в календаре была написана миграция в коде которой содержится создание таблицы с указанием наименования полей типа данных и правил заполнения.

```
$this->createTable('{{%event}}', [  
    'id' => $this->primaryKey(),  
    'idUser' => $this->integer()->notNull(),  
    'title' => $this->string(64)->notNull(),  
    'allDay' => $this->boolean(),  
    'start' => $this->dateTime()->notNull(),  
    'end' => $this->dateTime(),
```

```

        'description' => $this->text(),
        'created_at' => $this->integer(),
        'updated_at' => $this->integer()
    ], $tableOptions);

```

После чего в коде миграции прописывается связь между полем содержащим в себе информацию о личном номере пользователя который создал событие и таблицей пользователей, полем содержащим личный номер пользователя.

```

    $this->addForeignKey(
        'fk-event-idUser',
        '{{%event}}',
        'idUser',
        '{{%user}}',
        'id'
    );

```

2.2.4 Генерация кода средством Gii

Gii представлен в Yii2 как модуль. С помощью данного модуля были сгенерированы модели, контроллеры и представления для последующей работы с ними. Gii генерирует шаблон кода для последующего редактирования. Пример создание модели для таблицы задач представлен на рисунке 5.

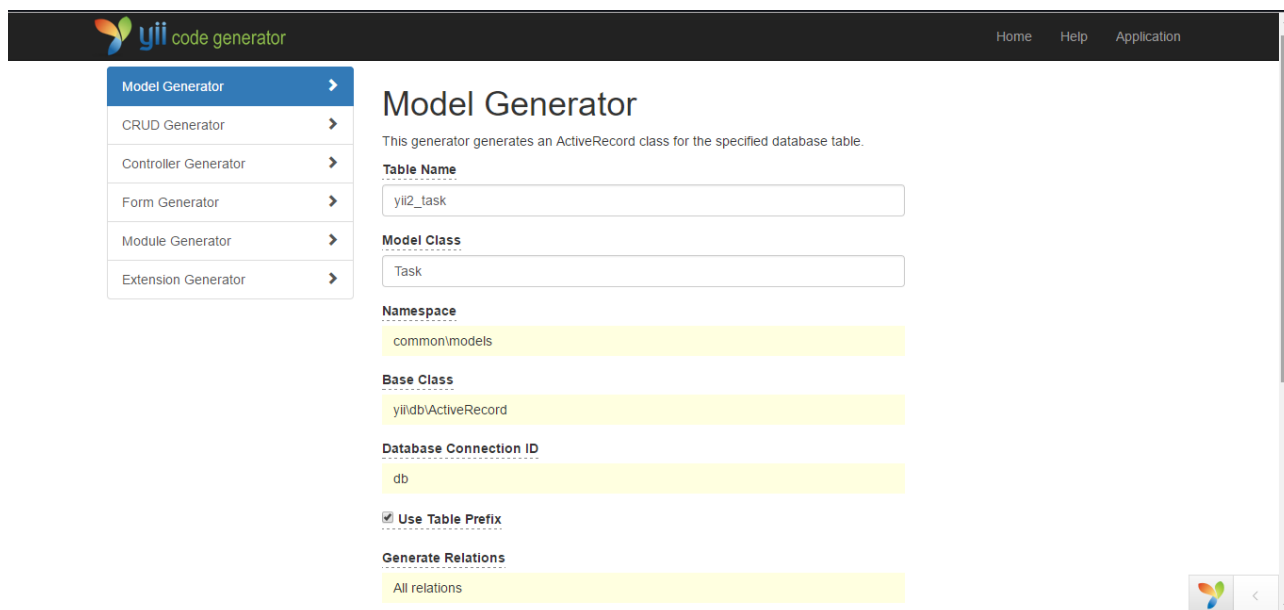


Рисунок 5 - Создание модели при помощи Gii

Для создание модели вызывается модуль Gii путём добавления в адресной строке браузера после доменного имени нашего проекта "\Gii". Входные данные необходимые для создания модели:

- имя таблицы;
- имя модели;
- адрес расположения файла модели;
- имя базового класса функции и другие дополнительные настройки.

2.2.5 Работа с фреймворком Bootstrap 3

Bootstrap 3 - Фреймворк позволяющий создавать адаптивные, современные, кросс-браузерные и стандартизированные интерфейсы. Данный Фреймворк отлично подходит для создания адаптивных шаблонов начиная от широкоэкранных мониторов и до экранов мобильных устройств с маленьким разрешением.

В файлах можно увидеть следующую структуру и содержание. Все файлы логически сгруппированы по общим свойствам и содержащие обе версии:

минимизированную и компилированную. На рисунке 6 приведена структура файлов Bootstrap 3.

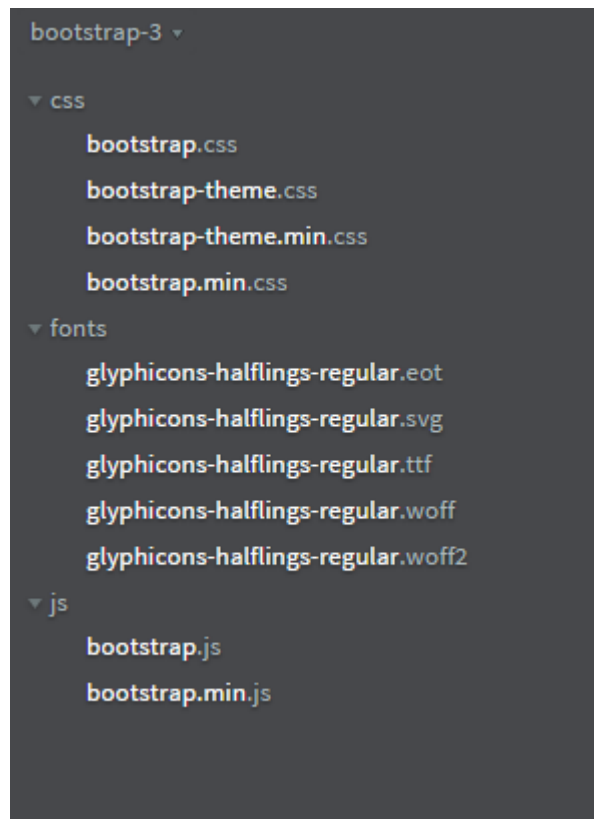


Рисунок 6 - Структура Bootstrap

Bootstrap 3 использовался при создании всего интерфейса web-приложения. Для размещения окна с полями ввода, для авторизации на сайте использовался следующий код.

```
<div class="col-lg-offset-4 col-lg-4 col-md-offset-4 col-md-4"
id="login_form__common">
```

Фрагмент этого кода отвечает за выделение на средних и больших мониторах 4 колонки из 12, с отступом слева 4 колонки, под окно авторизации в web-приложении.

Общий вид окна авторизации представлен на рисунке 7.

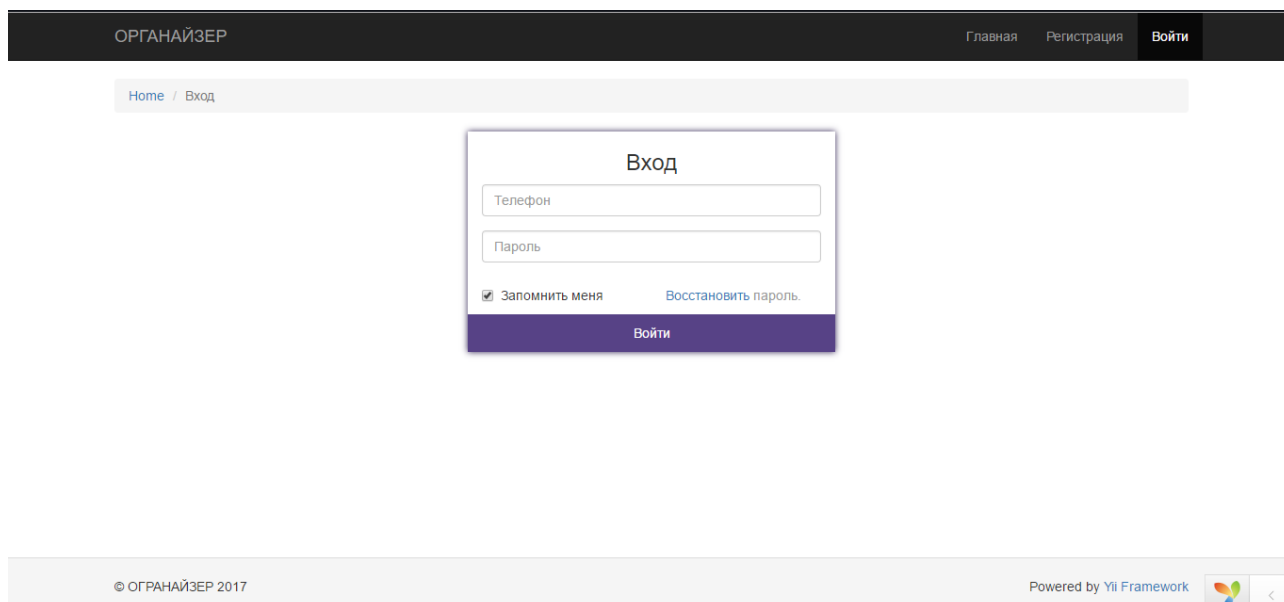


Рисунок 7 - Общий вид окна авторизации

2.3 Режимы работы web-приложения "Организер"

2.3.1 Регистрация пользователей

Для начала работы с web-приложением "Организер" пользователю необходимо пройти регистрацию. Для регистрации в приложении, указываются основные данные пользователя, его фамилия имя отчество, номер телефона, адрес электронной почты и пароль от учётной записи. После регистрации пользователь получает доступ к личному рабочему столу, возможности просматривать предложение о сотрудничестве от других компаний, а так же зарегистрировать в приложении собственную компанию.

Для обработки запросов на регистрацию пользователей в контроллере написана функция регистрации с следующим кодом.

```
public function actionSignup()  
{  
    $model = new SignupForm();  
    if (Yii::$app->request->isAjax) {  
        $model->load(Yii::$app->request->post());
```

```

        Yii::$app->response->format = \yii\web\Response::FORMAT_JSON;
        return ActiveForm::validate($model);
    }
    if ($model->load(Yii::$app->request->post())) {
        $model->validate();
        if ($user = $model->signup()) {
            if (Yii::$app->getUser()->login($user)) {
                return $this->goHome();
            }
        }
    }
    return $this->render('signup', [
        'model' => $model,
    ]);
}

```

При нажатии на кнопку регистрации вызывается новое представление, в которое подгружается форма регистрации нового пользователя, со всеми необходимыми полями. После заполнения данных в поля, происходит автоматическая проверка корректности введённых данных, вызывается функция валидации формы, которая проверяет наличие данных в полях отмеченных в модели как "необходимые для заполнения". После валидации и нажатия на кнопку "Зарегистрироваться" происходит отправка данных на контроллер, где все данные обрабатываются и отправляются в базу данных.

Общий вид окна регистрации представлен на рисунке 8.

ОРГАНАЙЗЕР

Главная Регистрация Войти

Home / Регистрация Organizer.ru

Регистрация Organizer.ru

Телефон

Фамилия

Имя Отчество

Email

Пароль

Повторите пароль

Зарегистрироваться

© ОГРАНАЙЗЕР 2017

Powered by Yii Framework

Рисунок 8 - Общий вид окна регистрации

2.3.2 Работа с компаниями

Для регистрации собственной компании пользователь указывает название компании, краткое описание компании и контактный телефон. После регистрации появляется пункт меню "компания" в котором директор компании может просматривать список сотрудников компании. В этом же разделе пользователь может отправлять предложения о работе или увольнять старых сотрудников.

Представление окна регистрации как и другие представления построены на основе фреймворка bootstrap 3.

Общий вид окна регистрации компании представлен на рисунке 9.

ОРГАНАЙЗЕР

ГлавнаяРабочий столПредложенияСоздать компаниюВыйти (+79992222222)

Home / Компания / Регистрация Компании

Регистрация Компании

Название компании

Описание

Телефон

Создать

© ОРГАНАЙЗЕР 2017

Powered by Yii Framework

Рисунок 9 - Общий вид окна регистрации компании

После заполнения всех полей в окне регистрации компании, данные отправляются на контроллер. Для обработки данных на контроллере была написана следующая функция.

```
public function actionCreate()  
{  
    $model = new Company();  
    $modelWork = new Worker();  
    $user = \Yii::$app->user->identity;  
    if ($model->load(\Yii::$app->request->post())) {  
        $model->user_created = $user->id;  
        if ($model->save()){  
            $modelWork->user_id = $user->id;  
            $modelWork->company_id = $model->id;  
            $modelWork->user_role = 'director';  
            $modelWork->save();  
            $user->role = $this->nextRoleUser;  
            if ($user->save()) {  
                return $this->redirect('/site/index');  
            }  
        }  
    }
```

```

    } else {
        return $this->render('create', [
            'model' => $model,
        ]);
    }
}

```

Первым шагом при вызове функции регистрации новой компании, происходит создание двух новых моделей, модели компании, и модели работника. После создания моделей происходит передача полученных данных из представления в модель компании, после чего модель компании сохраняется в базу данных. Следующим шагом происходит запись всех необходимых данных в модель работника, таких как:

- ID работника - личный номер сотрудника, в данном случае равен ID пользователя под аккаунтом которого происходила регистрация компании;
- ID компании - идентификатор компании в которую устроился работник;
- Должность сотрудника - в данном случае должность присваивается "директор".

После записи всех необходимых данных происходит сохранение в базу данных. Таким образом, при создании компании, в таблице "компании" появляется запись о зарегистрированной компании, а в таблице "работники" создаётся запись о первом работнике должность которого "директор".

Добавление новых пользователей в компанию происходит по приглашению, директор компании вводит данные человека, которого он хочет пригласить на работу в свою компанию, в базе данных создаётся запись с предложением о работе и данными о сотруднике. После чего, пользователь которому было отправлено приглашение на работу может либо принять его либо удалить. В случае если пользователь принимает приглашение, его статус меняется на тот, который ему предложили в приглашении и в базу данных заносятся данные о новом работнике компании. Для реализации обработки

данных о новом пользователе в компании в контроллере был прописан следующий код.

```
public function actionTakeOffer(){  
    $user = \Yii::$app->user->identity;  
    if($offer_id = \Yii::$app->request->post('offer_id')){  
        $offer = RequestCompanyUser::find()->where(['id' => $offer_id])->one();  
        if($offer){  
            $worker = new Worker();  
            $worker->company_id = $offer->company_id;  
            $worker->user_id = $user->id;  
            $worker->user_role = $offer->role_improver;  
            if($worker->save()){  
                $userChild = new UserChild();  
                $userChild->user_parent_id = $offer->user_request_id;  
                $userChild->user_child_id = $user->id;  
                if($userChild->save()){  
                    $user->role = $offer->role_improver;  
                    if($user->save()){  
                        if($offer->delete()) return $this->redirect('index');  
                    }  
                }  
            }  
        }  
    return 0;  
}
```

Выше представлен код, который реализует добавление в базу данных новой записи в таблицу "работники", после чего создаётся новая запись в таблице "начальники и работники" и меняется статус аккаунта пользователя.

2.3.3 Работа с задачами

Список задач - позволяет просматривать список составленных задач, присутствуют функции детального просмотра задач, удаление или изменение задач. Для аккаунта имеющего статус "директор", появляется функция назначение исполнителя для каждой из задач, исполнителя можно выбрать из списка сотрудников компании. Для аккаунта имеющего статус "работник" появляется дополнительный список, в которых директор компании назначил его исполнителем.

Пример отображения списка задач представлен на рисунке 10.

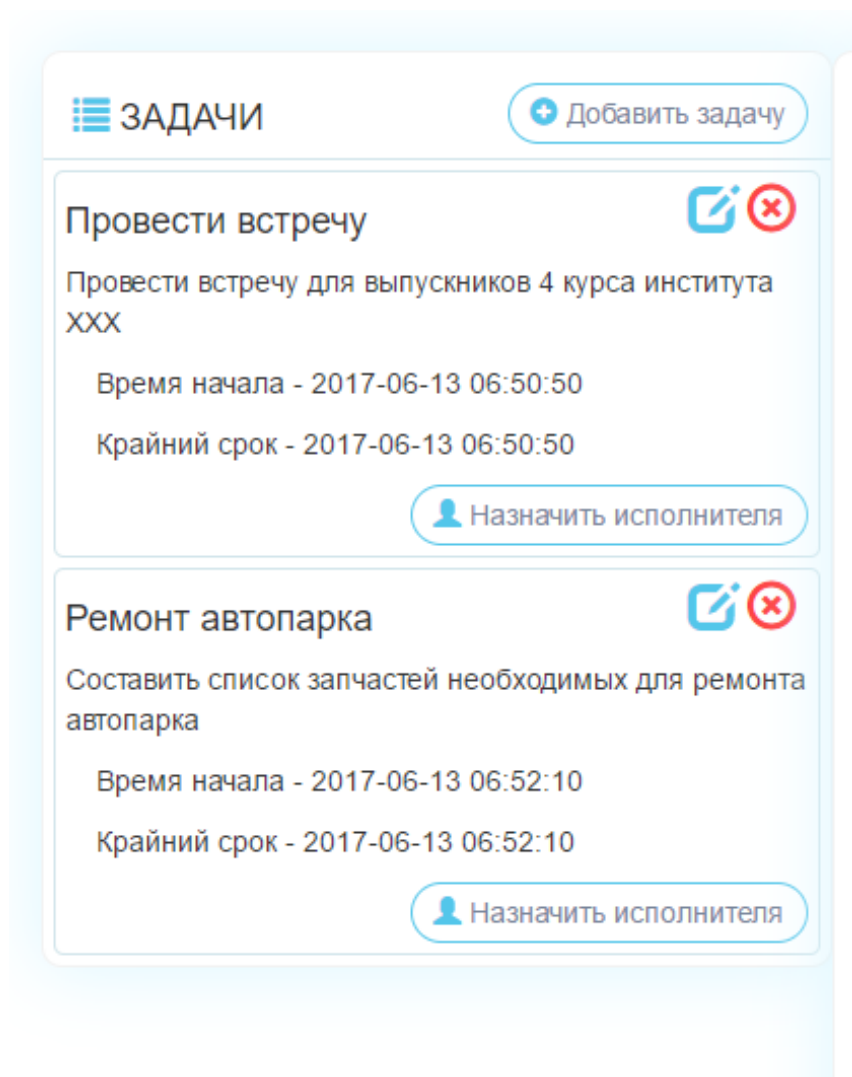


Рисунок 10 - Пример отображения списка задач

Для реализации функционала добавление новой задачи был написан следующий код.

```
public function actionCreate()  
{  
    $model = new Task();  
    $user = \Yii::$app->user->identity;  
    if (\Yii::$app->request->isAjax){  
        $model->load(\Yii::$app->request->post());  
        $model->user_create_id = $user->id;  
        if($model->save()) {  
            return $this->redirect(['/event/index', 'id' => $model->id]);  
        }  
    }  
}
```

Представленный выше код реализует получение данных из формы создание новой задачи и создаёт запись о новой задаче в базе данных.

Форма создания новой задачи подгружается в модальное окно, модальное окно написано при помощи средств фреймворка bootstrap 3.

Пример отображения формы создания новой задачи представлен на рисунке 11.

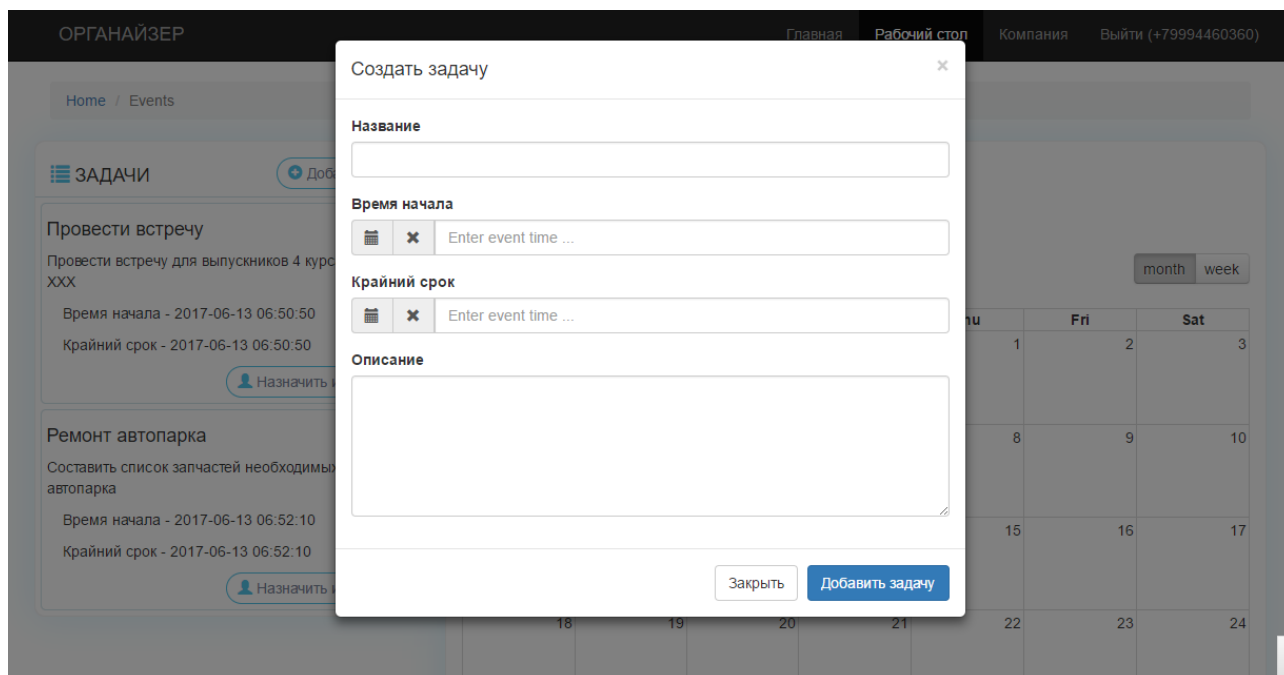


Рисунок 11 - Пример отображения формы создания новой задачи

2.3.4 Работа с календарём

Для реализации функционала позволяющего отображать календарь и обозначенные на нём события использовался готовый виджет "FullCalendar".

Форма добавления нового события на календарь так же как и при создании новой задачи отображается в модальном окне.

Календарь имеет два режима отображения событий "отображать месяц" и "отображать неделю".

Пример отображения задач на календаре в режиме "отображать месяц" приведён на рисунке 12.

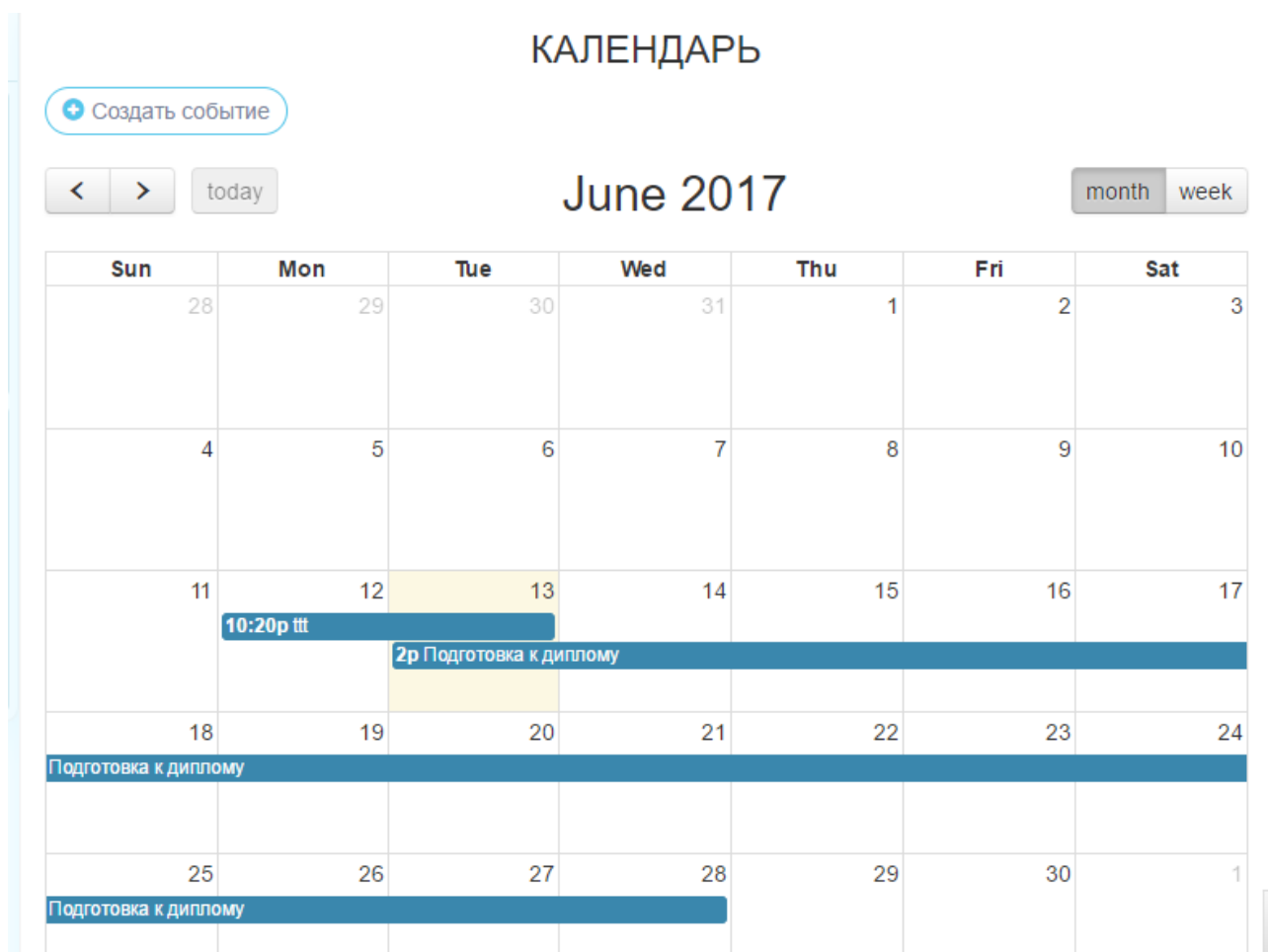


Рисунок 12 - Пример отображения задач на календаре в режиме "отображать месяц"

Пример отображения задач на календаре в режиме "отображать неделю" представлен на рисунке 13.

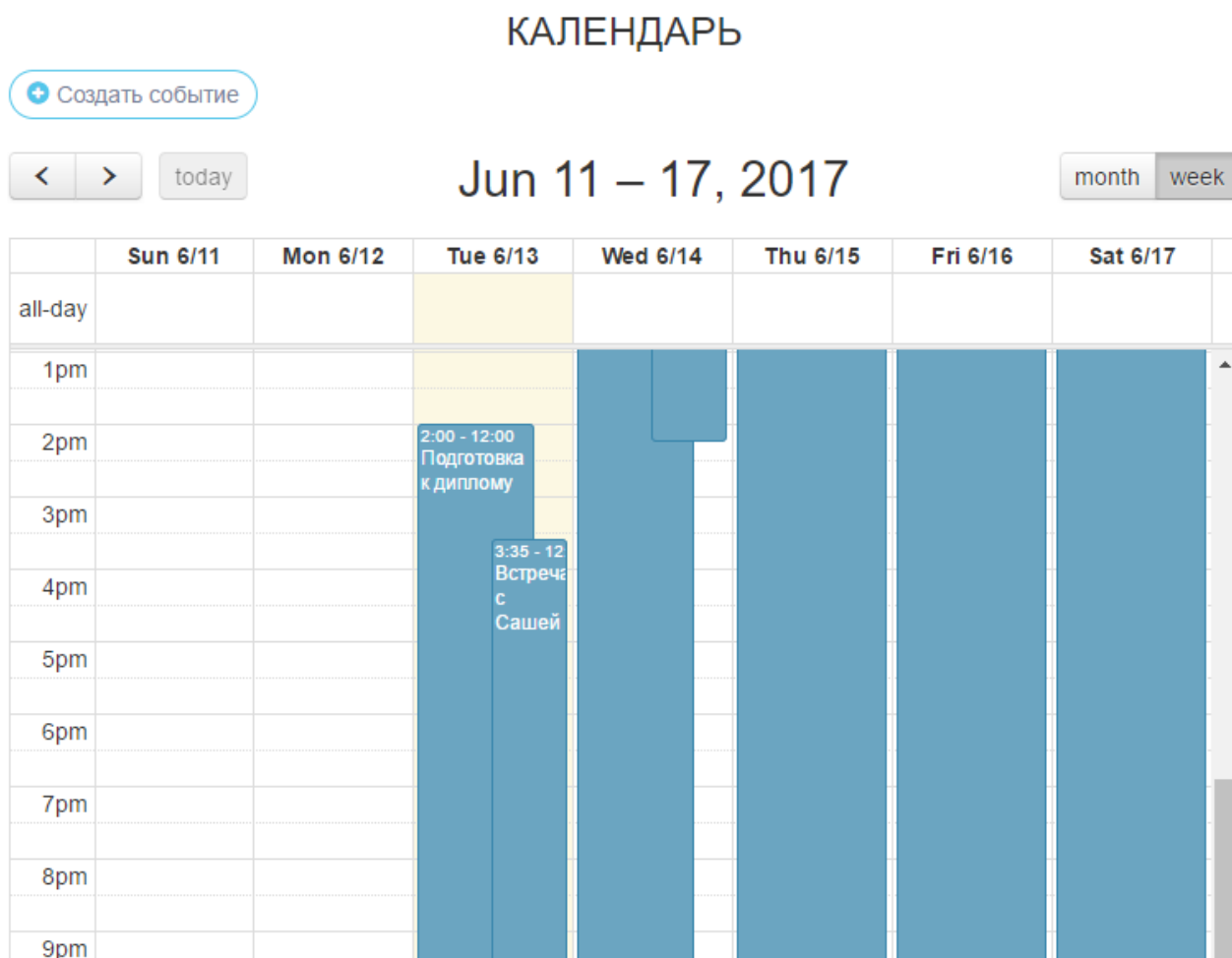


Рисунок 13 - Пример отображения задач на календаре в режиме "отображать неделю"

2.3.5 Распределение возможностей и привилегий

В приложении реализовано четыре типа учётных записей, основное разбитие возможностей пользователей происходит в главном меню. В зависимости от типа учётной записи в главном меню будут отображаться только пункты меню доступные данному типу аккаунта. Для реализации главного меню с проверкой типа учётной записи, был написан следующий код.

```
<?php
if (!\Yii::$app->user->isGuest){
    $user = \Yii::$app->user->identity;
    $user_privilege = $user->getRole();
```



```

if($user_privilege === 'director'){
    $label_company = ['label' => 'Компания', 'url' => ['/company/index']];
    $label_request_company_user = '';
    $label_desktop = ['label' => 'Рабочий стол', 'url' => ['/event/index']];
}
if($user_privilege === 'individual'){
    $label_company = ['label' => 'Создать компанию', 'url' => ['/company/create']];
    $label_request_company_user = ['label' => 'Предложения', 'url' =>
['/requestcompanyuser/index']];
    $label_desktop = ['label' => 'Рабочий стол', 'url' => ['/event/index']];
}
if($user_privilege === 'secretary'){
    $label_company = '';
    $label_request_company_user = '';
    $label_desktop = ['label' => 'Рабочий стол', 'url' => ['/event/index']];
}
if($user_privilege === 'worker'){
    $label_company = ['label' => 'Компания', 'url' => ['/company/index']];
    $label_request_company_user = '';
    $label_desktop = ['label' => 'Рабочий стол', 'url' => ['/event/index']];
}
}else{
    $label_company = '';
    $label_desktop = '';
    $label_request_company_user = '';
}
?>

```

ЗАКЛЮЧЕНИЕ

Применение разработанного программного комплекса для решения задач оперативного планирования рабочего времени и распределения задач позволило оптимизировать работу как отдельного пользователя, так и целой компании, создав единое общее оперативное пространство.

Разработана простая и удобная программа-органайзер. Которая поможет составить список дел на день/неделю/месяц, обозначить список задач, распределить задачи между сотрудниками.

Программа реализована в виде сетевого приложения, следовательно повышается мобильность пользователей и доступность использования.

Сетевое решение, предназначенное для обеспечения обмена данными и использования общих ресурсов компании должно быть открытым – должно поддерживать множество разнородных аппаратных средств, сетевых протоколов. Залогом эффективности его работы является ориентация на применение открытых, кроссплатформенных технологий.

Основным преимуществом данного приложения по сравнению с аналогами является отсутствие долгих и трудоёмких настроек перед началом работы, абсолютно бесплатное использование, не содержит избыточного функционала.

Полученные результаты могут быть использованы для оперативного принятия решения о выполнении поступивших задач.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Википедия [Электронный ресурс] // Свободная интернет энциклопедия. – Режим доступа: <https://ru.wikipedia.org/wiki/>
2. Основы HTML [Электронный ресурс] // Электронная документация. – Режим доступа: <http://html5book.ru/osnovy-html/>
3. Основы CSS [Электронный ресурс] // Электронная документация. – Режим доступа: <http://html5book.ru/osnovy-css/>
4. Основы JavaScript [Электронный ресурс] // Электронная документация. – Режим доступа: <http://html5book.ru/osnovy-javascript/>
5. Руководство Yii 2.0 [Электронный ресурс] // Электронная документация. – Режим доступа: <http://www.yiiframework.com/doc-2.0/guide-index.html>
6. Built with Bootstrap [Электронный ресурс] // Официальный сайт. – Режим доступа: <http://getbootstrap.com/>